

# MQTT 消息队列压力测试。

作者: [a929569603](#)

原文链接: <https://ld246.com/article/1570974390604>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

最近项目中对MQTT消息队列进行压力测试，简单记载MQTT消息插件的使用。

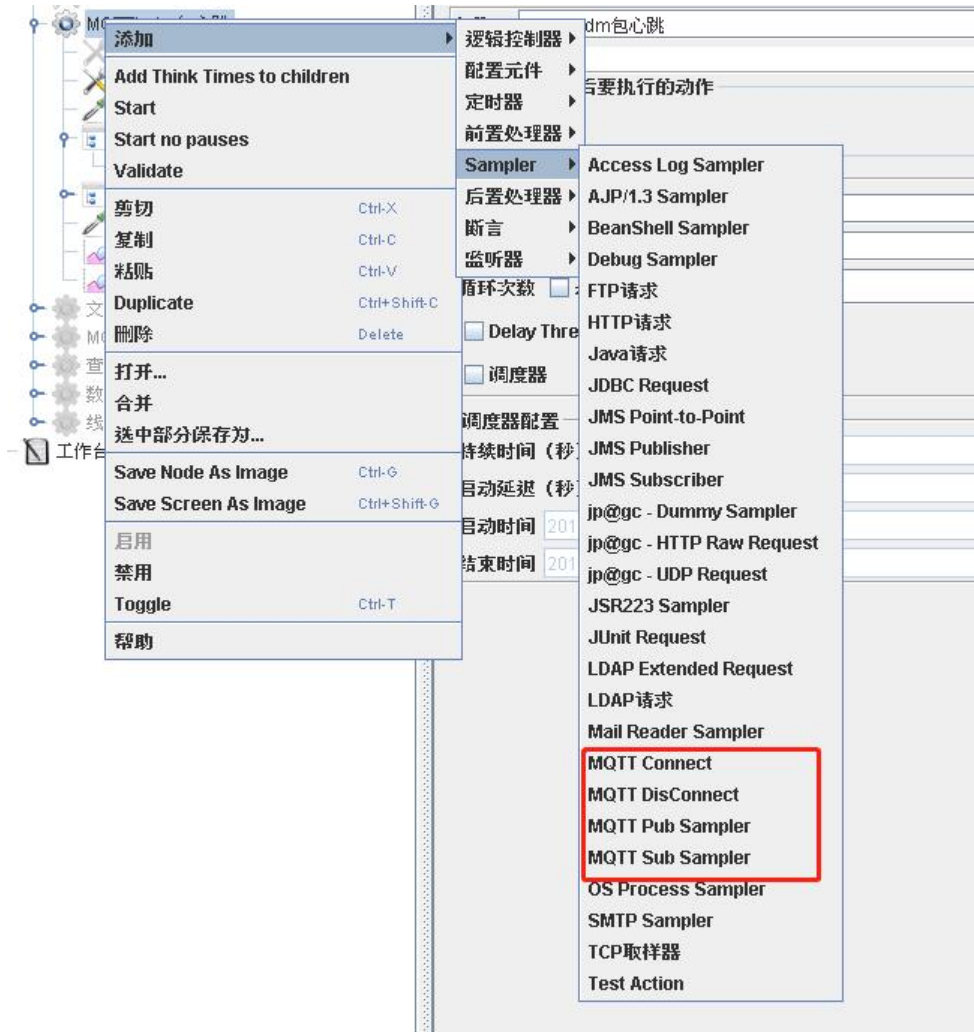
## MQTT消息队列压力测试

环境准备：

jmeter插件下载：

[mqttxmeter1.0.1jarwithdependencies.jar](#)

把MQTT插件放在 %JMeter\_Home%/lib/ext下。重启jmeter。



MQTT连接。

**Server name or IP:** 被测MQTT服务器地址。

**Port number:** TCP连接的端口1883, SSL连接则是8883。

**Timeout(s):** 连接超时设置，以秒为单位。

**user name:** 连接MQTT的帐户名。

**password:** 连接MQTT的密码。（具体询问开发）

**ClientId:** 客户端标识，具体询问开发被模拟的客户端标识（**注意一个标识快速连接多次会连接失败，发记得参数化（经验）**）。

**Keep alive(s):** 心跳信号发送间隔。例如，300表示客户端每隔300秒向服务器发出ping请求，以保持连接活跃。（**注意，不正常disconnect，连接还会继续保持。除非关掉jmeter. 正常mqtt disconnect请无视。**）

**Connect attempt max:** 第一次连接失败后，尝试重连的最大次数。超过该次数则认为连接失败。

**Reconnect attempt max:** 后续连接过程中连接失败后，尝试重连的最大次数。超过该次数则认为连接失败。

好了，连接上就可以进行消息发布了。

## MQTT发布

```

8  }
9  },
10 "deviceCodeList": ["200319090000${deviceCode}"],
11 "deviceTypeCode": "2003",
12 "merchantId": "1",
13 "requestId": "a8c65f5fa869442d90f56b5c01fe7b0d",
14 "storeCode": "${storeCode}",
15 "timeStamp": "1526196582885",
16 "type": "2"
17 }

```

**QoS Level:** 服务质量，取值为0, 1, 2, 分别代表MQTT协议规范里的至多一次 (AT\_MOST\_ONCE, 至少一次 (AT\_LEAST\_ONCE), 精确一次 (EXACTLY\_ONCE) (网上COPY, 具体我就用了0, 用

2发送失败。)

**Topic name:** 做为发布方, 把消息发布到所属的话题中。

**Add timestamp in payload:** 如果勾选, 发布的消息体开头会附带当前时间戳, 利用它可以在消息收端计算消息达到的延时。不勾选则只发送实际的消息体。

1. **Message type:** 目前支持三种消息类 (我只用了String,其它请看网上的其它介绍。)

1. String: 普通字符串 (如上面截图所示, 你要发送的消息。具体上报什么, 可找开发要相关json。)

2. Hex String: 以16进制数值表示的串, 比如字符串Hello, 可以表示为48656C6C66 (其中, 48在ascii表中对应字母H, 依次类推)。通常16进制串用来构造非文本的消息体, 例如描述某些私有的协议交互和控制信息等等。

3. Random string with fixed length\*: 按指定长度生成随机的串作为消息体。

此时消息发送成功了。

## 消息订阅



**topic name(s):** 想要订阅的主题。(可以订阅多个) 多个主题以英文逗号分隔。

**Sample on:**

1. **specified elapsed time(ms):** 订阅的主题每多少毫秒作为一个结果展示(个人理解)。在这些毫秒内所收到的消息全部展示在一个结果内。

2. **number of received messages:** 每收到多少个消息作为一个结果展示(个人理解)。

## 断开连接



注意: 主动分支MQTT DisConnect可以断开。若测试过程中手动停止, 致使线程未运行DisConnect, 则连接不会断开。(已经过验证。)

## 测试过程中遇到的问题:

1. MQTT连接数未修改。连接到1W报错连接失败。
2. 修改MQTT连接数后，成功并发到5W，后续还会再进行测试。

```
## listener.max_connections is an integer or
## the maximum number of concurrent connection
## on the protocol level by:
## - listener.tcp.max_connections
## - listener.ssl.max_connections
## - listener.ws.max_connections
## - listener.wss.max_connections
## or on the listener level by:
## - listener.tcp.my_tcp_listener.max_connecti
## - listener.ssl.my_ssl_listener.max_connecti
## - listener.ws.my_ws_listener.max_connection
## - listener.wss.my_wss_listener.max_connecti
##
## Default: 10000
##
## Acceptable values:
## - an integer
## - the text "infinity"
listener.max_connections = 100000
```

